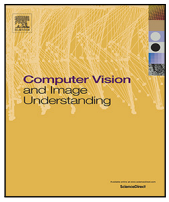
Computer Vision and Image Understanding 234 (2023) 103743

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu



Unsupervised domain adaptation for semantic segmentation via cross-region alignment

Zhijie Wang [a], Xing Liu [a], Masanori Suganuma [a,b], Takayuki Okatani [a,b,*]

[a] Graduate School of Information Sciences, Tohoku University, Japan
[b] RIKEN Center for AIP, Japan

ARTICLE INFO

Communicated by Nikos Paragios

Keywords:
Domain adaptation
Semantic segmentation
CNN

ABSTRACT

Semantic segmentation requires a lot of training data, which necessitates costly annotation. There have been many studies on unsupervised domain adaptation (UDA) from one domain to another, e.g., from computer graphics to real images. However, there is still a gap in accuracy between UDA and supervised training on native domain data. It is arguably attributable to the class-level misalignment between the source and target domain data. To cope with this, we propose a method that applies adversarial training to *align two feature distributions in the target domain*. It uses a self-training framework to split the image into two regions (i.e., trusted and untrusted), which form two distributions to align in the feature space. We term this approach *cross-region adaptation* (CRA) to distinguish it from the previous methods of aligning different domain distributions, which we call *cross-domain adaptation* (CDA). CRA can be applied after any CDA method. Experimental results show that this always improves the accuracy of the combined CDA method.

1. Introduction

Semantic image segmentation is one of the fundamental problems of computer vision (Minaee et al., 2021). Methods employing neural networks have achieved great success for the problem, which presume the availability of a large amount of labeled data. As manual annotation is costly, researchers have considered using synthetic images generated by computer graphics, for which precise pixel-level annotation is readily available (Richter et al., 2016; Ros et al., 2016; Wrenninge and Unger, 2018).

However, it is generally hard to apply neural networks trained with synthetic data to real images because of the distributional difference between synthetic and real images. Many studies have been conducted on domain adaptation (Long et al., 2015a; Sun and Saenko, 2016) to cope with the difference known as domain shift. Among several problem settings, the one that attracts researchers' most interest is unsupervised domain adaptation (UDA) (Saito et al., 2018; Zhou et al., 2022; Pasqualino et al., 2022). It is to train a model using labeled data in a domain (called the source domain) so that it will work well on data in a different domain (called a target domain) for which labels are not available.

There are currently two approaches to UDA for semantic segmentation. One is *adversarial training* (Tsai et al., 2018; Vu et al., 2019a,b; Wang et al., 2020; Bucher et al., 2021), which attempts to obtain domain-invariant features by aligning the data distributions of the source and target domains. An issue with this approach is that while it may be easy to align the two distributions as a whole, it is hard to attain class-level alignment, leading to suboptimal results. The other approach is *self-training*, in which a teacher model trained with the labeled data in the source domain is used to generate pseudo labels of the target domain data and use them for training a student model (Zou et al., 2018, 2019; Li et al., 2019). An issue with this approach is that pseudo labels could be inaccurate, which will lead to unsatisfactory performance.

A promising direction for further improvements is to *integrate* adversarial training and self-training, as is attempted by recent studies (Mei et al., 2020). This paper proposes a new approach in the same direction, which applies adversarial training to align two feature distributions *in* the target domain. Using a self-training framework, it splits target domain images into two regions, thereby specifying the feature distributions to align.

We refer to this approach as *cross-region adaptation* (CRA) to differentiate it from the conventional method of aligning distributions across different domains, which we will call *cross-domain adaptation* (CDA). The objective of CRA is to address class-level misalignment between the source and target domain data, as illustrated in Fig. 1. CRA is intended to be used as an add-on to an existing UDA method. Our experimental results on three benchmark tasks, GTA5 → Cityscapes, SYNTHIA → Cityscapes, and Synscapes → Cityscapes, demonstrate that

* Corresponding author at: Graduate School of Information Sciences, Tohoku University, Japan.
E-mail address: okatani@vision.is.tohoku.ac.jp (T. Okatani).

[1] The code used for our experiments is found in https://github.com/zhijiew/CRA.

https://doi.org/10.1016/j.cviu.2023.103743
Received 14 October 2022; Received in revised form 24 May 2023; Accepted 26 May 2023
Available online 9 June 2023

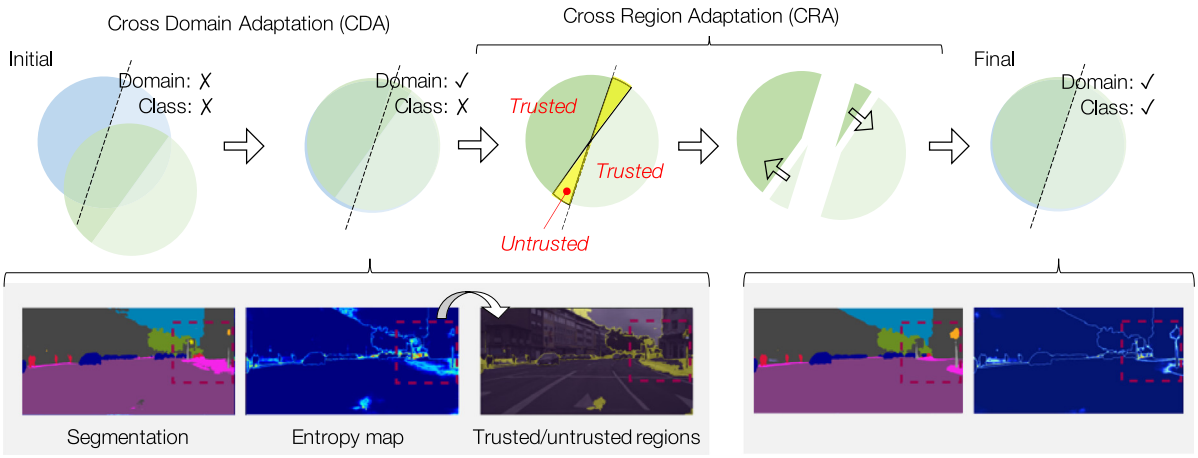


Fig. 1. Illustration of our method. Conventional cross-domain adaptation (CDA) aims to align the feature distributions of different domains, but it may not address differences in their classes. Our proposed cross-region adaptation (CRA) uses a confidence map to divide an image into trusted and untrusted regions. The feature distributions of these two regions are then aligned using adversarial training.

Table 1
The proposed method, called cross-region adaptation (CRA), is applied after any existing UDA method and consistently leads to improved performance. Results for UDA from GTA5 → Cityscapes are shown.

Method	Baseline	+CRA	Δ
ASN (Tsai et al., 2018)	42.4	43.4	+1.0
ADVENT (Vu et al., 2019a)	43.8	46.7	+2.9
IntraDA (Pan et al., 2020)	46.3	47.0	+0.7
FADA (Wang et al., 2020)	50.1	52.2	+2.1
IAST (Mei et al., 2020)	52.2	54.1	+1.9
ProDA (Zhang et al., 2021)	57.5	58.6	+1.1
DAFormer (Hoyer et al., 2022a)	54.7	55.5	+0.8
MIC (Hoyer et al., 2023)	58.2	59.2	+1.0

CRA consistently enhances the accuracy of the base UDA method.[1] We summarize our findings in Table 1.

2. Related work

2.1. Semantic segmentation

Methods based on convolutional neural networks (CNNs) have been the most successful for semantic segmentation. FCN is the first fully convolutional network for the pixel-level classification task proposed in a pioneering work (Long et al., 2015b). Later, UNet (Ronneberger et al., 2015) and SegNet (Badrinarayanan et al., 2017) were proposed, which are networks consisting of an encoder and a decoder, leading to better performance. Architectural designs have been extensively studied since then such as DeepLab (Chen et al., 2017), PSPNet (Zhao et al., 2017), UperNet (Xiao et al., 2018), ICTNet (Chatterjee and Poullis, 2021), and RFCNet (Zhang and Aliaga, 2022), to name a few.

2.2. Unsupervised domain adaptation

There are two approaches to unsupervised domain adaptation (UDA) for semantic segmentation, i.e., adversarial training and self-training. The former mainly attempts to decrease a domain gap by performing adversarial training in the feature space (Dong et al., 2020; Wang et al., 2020), in the input space (Gong et al., 2019; Lee et al., 2018), or in the output space (Tsai et al., 2018). The core idea of self-training is to generate pseudo labels for target domain samples and use them for training the model (Chen et al., 2019). CBST (Zou et al., 2018) and CRST (Zou et al., 2019) conduct class-balanced self-training and confidence-regularized self-training, respectively, to generate better pseudo labels.

Several attempts have also been made to combine adversarial training and self-training for improving performance (Li et al., 2019; Zheng and Yang, 2020; Mei et al., 2020). BLF (Li et al., 2019) uses pseudo-labels without filtering, which can result in label errors. AdaptMR (Zheng and Yang, 2020) filters pseudo-labels but disregards the filtered-out pixels, which may contain valuable information. IAST (Mei et al., 2020) utilizes filtered pseudo-labels for supervised training and applies entropy minimization, a semi-supervised learning method, to the filtered-out pixels. In contrast, our method splits target-domain images into trusted and untrusted regions based on an entropy-based confidence map and applies adversarial training to the features of these two regions. This approach aims to achieve finer alignment of the source- and target-domain distributions. Consequently, our method is distinct from the aforementioned methods, and experimental results demonstrate its superior performance.

Further studies in orthogonal directions to the above have been conducted, improving the performance of UDA. DPL (Cheng et al., 2021) proposes dual path learning framework consisting of two complementary and interactive single-domain adaptation pipelines aligned in the source and target domain, respectively, to promote each other in an interactive manner. DAFormer (Hoyer et al., 2022a) generates the target pseudo labels by a dynamically updated teacher model, whose weights are set as the exponentially moving average (EMA) of the student model. DAFormer stabilizes training and avoids overfitting to the source domain via the rare class sampling, thing-class ImageNet feature distance, and learning rate warmup. HRDA (Hoyer et al., 2022b) presents a multi-resolution training approach for the UDA in semantic segmentation, combining high-resolution and low-resolution images to capture both fine segmentation details and long-range context dependencies, respectively. DecoupleNet (Lai et al., 2022) decouples the feature distribution alignment task and the segmentation task to focus more on the segmentation task. It introduces self-discrimination (SD) and online enhanced self-training (OEST) components to improve the quality of pseudo labels and target domain feature learning. MIC (Hoyer et al., 2023) proposes a masked image consistency module to enforce the consistency between predictions of masked target images and pseudo-labels generated by an EMA teacher, the network learns to infer the predictions of the masked regions from their context. The proposed CRA is designed to be used in conjunction with a base UDA method. It is agnostic to the choice of base method, as demonstrated in Table 1.

2.3. Use of predictive uncertainty for segmentation

Many methods utilize the entropy of predicted class probabilities to measure the uncertainty of the model's prediction for better training.

ADVENT (Vu et al., 2019a) proposes to use entropy for adversarial training and also for unsupervised training on target domain data (i.e., entropy-minimization). DADA (Vu et al., 2019b) estimates the scene depth from the same input images at training time. It aligns the source and target distributions in the standard feature space and jointly in the depth space, aiming for a more accurate alignment. ESL (Saporta et al., 2020) uses the entropy to assess the confidence of the prediction better and filter pseudo labels for self-training while ignoring unselected pixels. IntraDA (Pan et al., 2020) splits target domain data into easy and hard samples based on the entropy and performs intra-domain adversarial training. Although our method is similar in using adversarial training within the target domain, ours consider aligning features from different image regions; more importantly, its performance is much higher.

3. Proposed method

3.1. Revisiting adversarial domain adaptation

Before explaining our method, we revisit the adversarial training for conventional cross-domain adaptation (CDA). The problem is stated as follows. We are given labeled data $X_s = \{(x^{(s)}, y^{(s)})\}$ of the source domain and unlabeled data $X_t = \{x^{(t)}\}$ of the target domain. We assume here the two domains share the same K semantic classes to predict. We wish to train a segmentation network $G = C \circ F$, where C is a classifier and F is a feature extractor. We first train G on X_s by minimizing the cross-entropy loss:

$$\mathcal{L}_{seg}^{cda} = -\sum_{i=1}^{H\times W}\sum_{k=1}^{K} y_{ik}^{(s)} \log p_{ik}^{(s)}, \tag{1}$$

where $p_{ik}^{(s)} = G(x^{(s)})$ is the softmax probability of pixel i belonging to class k and $y_{ik}^{(s)}$ is the ground-truth one-hot label.

To make G work well also with the target domain images, we consider a discriminator D that distinguishes the domain (i.e., source or target) from an input feature $f = F(x)$. Freezing C, we then train F and D in an adversarial fashion, aiming at aligning the distributions of the two domains in the feature space.

3.2. Cross region adaptation (CRA)

Successful cross-domain adaptation (CDA) should result in well-aligned source and target distributions. However, this does not guarantee class-level alignment between the domains, as recognized by the community (Wang et al., 2020). This can be the case even for UDA methods that do not rely on adversarial training. Therefore, our goal is to address any remaining class-level misalignment that may exist after applying a base UDA method.

3.2.1. Outline of the method

After applying a base UDA method and obtaining the segmentation network G, inaccuracies in class-level alignment can result in errors near class boundaries in the feature space, as demonstrated in Fig. 1. Our objective is to decrease the number of misclassified pixels, or in other words, to reposition the pixels currently situated on the wrong side of the class boundary to the correct side within the feature space.

Erroneous pixels can be identified by analyzing the uncertainty of the class prediction. In this study, we use the entropy of the predicted class probability for this purpose. To be more precise, we initially classify each image pixel into two categories, namely, *trusted* and *untrusted*, by setting a threshold for the entropy. Further information on this process can be found in Section 3.3.

Next, we align the feature distributions of the trusted and untrusted pixels by conducting adversarial training on these distributions within the target domain. Unlike CDA that aligns feature distributions of source- and target-domain images, CRA aligns the distributions of trusted and untrusted pixels specifically within the target domain. Alongside this adversarial training, we use standard self-training by training the network with pseudo-labels on trusted pixels, which were provided by the model's earlier version.

3.2.2. Assumptions and analysis (why CRA works?)

If the two following conditions are met, the aforementioned procedure should successfully decrease the number of misclassified pixels:

- All trusted pixels are classified correctly, and some of the untrusted pixels are misclassified.
- The number of untrusted pixels is relatively small, as illustrated in Fig. 3.

If the first condition is satisfied, CRA endeavors to align untrusted pixels, which may be misclassified, with the trusted pixels that are classified correctly in the feature space, thus achieving the desired outcome. However, in practice, aligning these two distributions may produce a suboptimal result. For instance, it may alter the class boundary, which can result in more misclassifications. However, this is not the case when the second condition is met. With a limited number of untrusted pixels, their alignment with the trusted ones will have minimal impact on the class boundary, which is strongly supported by many trusted pixels.

While we cannot guarantee that the above assumptions hold precisely, it is reasonable to assume that they mostly hold after applying a base UDA method with reasonably good performance. Then, experimental validation is necessary to determine the degree to which these assumptions must hold for CRA to be effective. We will present the results of extensive experiments conducted to investigate this.

3.3. Details of the method

3.3.1. Overall procedure

The model (G) that performs well on the target domain is obtained through four steps using the labeled source domain data X_s and unlabeled target domain data X_t. Firstly, G is trained on X_s in a standard supervised fashion. Secondly, UDA is performed with any existing method to fine-tune G and train D as shown in Step 1 of Fig. 2. Next, pseudo labels and confidence maps are generated for the images in X_t, and they are divided into trusted and untrusted regions based on the confidence maps, as described in Section 3.3. This is shown in Step 2 of Fig. 2. Finally, CRA is applied between the trusted and untrusted regions to fine-tune G and D, as explained above, shown in Step 3 of Fig. 2.

3.3.2. Choosing trusted and untrusted image regions

As outlined above, in the second step, a segmentation network G is obtained after the application of an existing UDA method. Next, we classify each pixel of the image into either *trusted* or *untrusted* using the entropy of the predicted class probability. Specifically, we first apply G to each image $x^{(t)}$ of the target domain, obtaining the softmax probability $p^{(t)} = [p_{i1}^{(t)}, \dots, p_{iK}^{(t)}]$ at pixel i of $x^{(t)}$. We calculate the entropy of the class probability as

$$e_i = -\frac{1}{K \log K}\sum_{k'=1}^{K} p_{ik'}^{(t)} \log(p_{ik'}^{(t)}). \tag{2}$$

We then classify each image pixel (i) by thresholding e_i with a constant λ into *trusted* or *untrusted*. Let m_i be a pixel-wise mask indicating the pixel being trusted, which is given by:

$$m_i = \begin{cases} 1 & \text{if } e_i < \lambda \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The mask for untrusted pixels is obtained as $\overline{m}_i = 1 - m_i$.

The choice of the hyperparameter λ is crucial. If validation data are available, we should select λ using them, as assumed in previous studies. Alternatively, we can use a simple method to calculate a good value for λ. The minimum value for the entropy is obtained when a single class has a probability of 1, and the theoretical maximum is obtained when all class probabilities are equal, i.e., $p_k = 1/K$ ($k = 1, \dots, K$), where K is the number of classes. In the case where

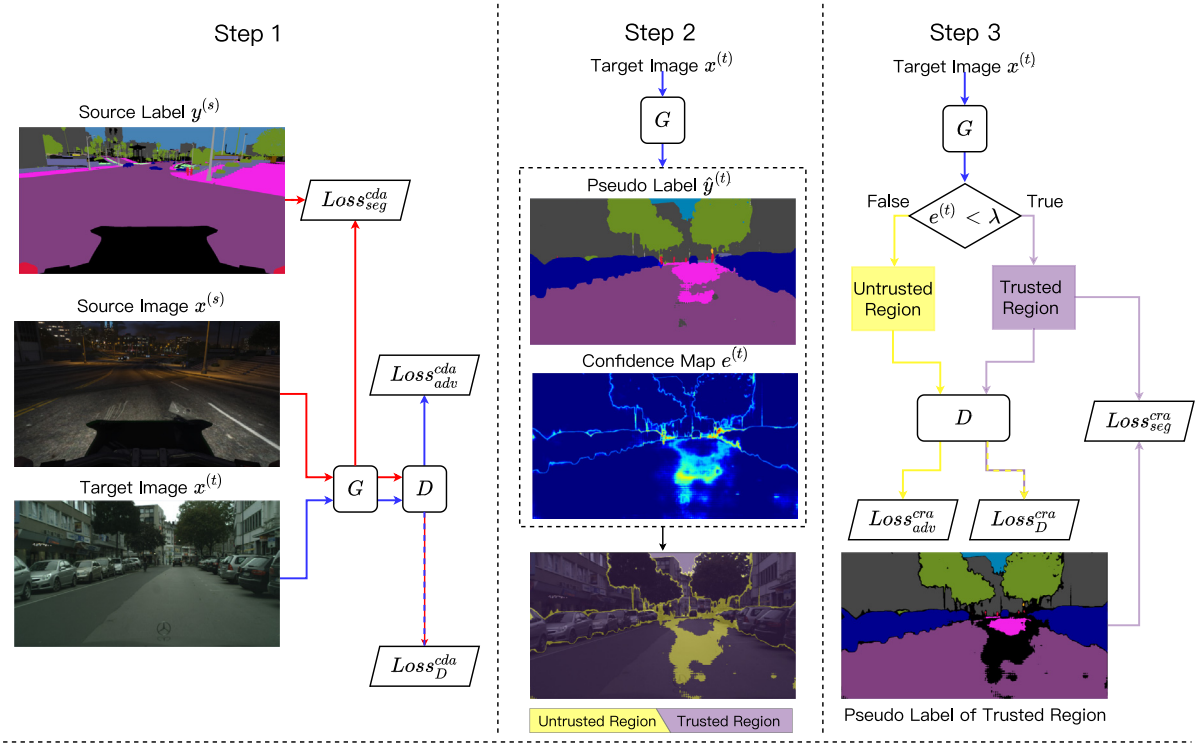


Fig. 2. Overall procedure of training a model (G) using labeled source domain data $\{(x^{(s)}, y^{(s)})\}$ and unlabeled target domain data $\{x^{(t)}\}$. After training G on the source domain data, we first apply a CDA method to align the feature distributions of the two domains, yielding updated G and D (Step 1). Next, for the target domain data, we generate pseudo labels $\hat{y}^{(t)}$ and split each image into trusted and untrusted regions based on a confidence map $e^{(t)}$ (Step 2). We finally apply the proposed CRA training to align the feature distributions of the two regions within the target domain data, resulting in updated G (and D).

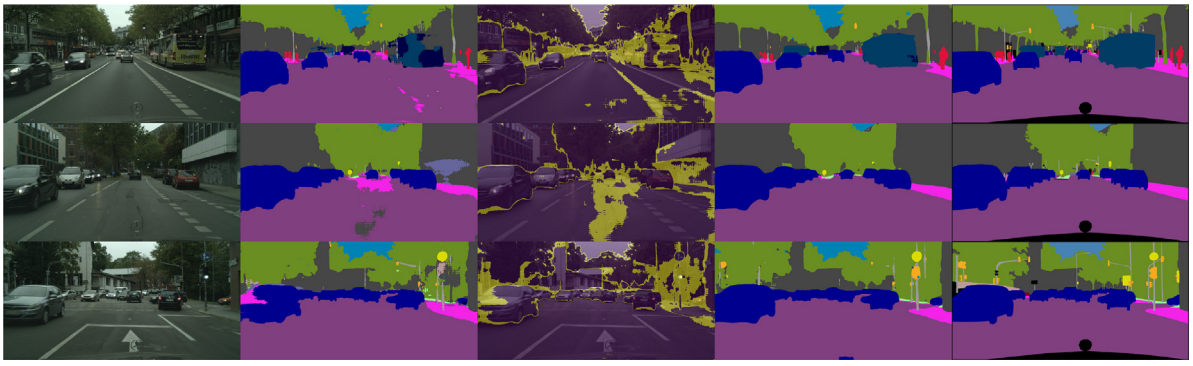


Fig. 3. Examples of the results on GTA5 → Cityscapes. From left to right, target images, results of a base method, FADA (Wang et al., 2020), trusted and untrusted regions (in purple and yellow, respectively), results of CRA, and ground truths. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the prediction is evenly split among multiple classes and we cannot choose a particular class, the minimum entropy is obtained when two classes have a probability of $1/2$ and the others have zero. This value is $\log 2/(K \log K)$. For the Cityscapes dataset (Cordts et al., 2016), which has $K = 19$ classes, it is 0.012. The threshold λ should be lower than this value, and we choose $\lambda = 0.01$ in our experiments. We will discuss the effectiveness of this approach in Section 4.9 again.

In segmentation data, class imbalance is always present. Classes that account for less than a few percent of the data tend to produce high entropy, and their pixels are usually classified as untrusted. To prevent this, it is necessary to adaptively adjust the threshold λ for these classes. However, we found a simple alternative that works well: multiplying the entropy of rare classes by a constant F_s. We identify the rare classes by counting the number of pseudo labels $\hat{y}^{(t)}$ (explained below) and thresholding it with $1/100$ of all pixels. In Section 4.7, we will discuss the effect of the entropy scaling factor F_s.

We will also use a pseudo label for each pixel of a target domain image $x^{(t)}$. We define this as

$$\hat{y}_{ik}^{(t)} = \begin{cases} 1 & \text{if } k = \arg\max_{k'} p_{ik'}^{(t)} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

3.3.3. Training of the network

We train the segmentation network $G = C \circ F$ and the discriminator D by fine-tuning the model trained in the previous step. We first train G with the trusted pixels of the target domain images using the above pseudo label $\hat{y}_{ik}^{(t)}$. We ignore the untrusted pixels here. We use the standard cross-entropy loss:

$$\mathcal{L}_{seg}^{cra} = -\sum_{i=1}^{H \times W} \sum_{k=1}^{K} m_i \hat{y}_{ik}^{(t)} \log p_{ik}^{(t)}. \tag{5}$$

The insertion of the mask m_i ensures the loss is computed over only the trusted pixels.

After this, we train G and D in an adversarial fashion, where G and D are updated alternately as follows. We first train G so that D will misclassify untrusted pixels as trusted pixels. We employ the fine-grained adversarial approach (Wang et al., 2020) for the training of G and D, in which we first generate a domain encoding label containing class probabilities, i.e., $[a; \mathbf{0}]$ for source domain and $[\mathbf{0}; a]$ for target domain, where a is a K-vector containing the class probabilities and $\mathbf{0}$ is a zero vector of size K. These are computed from the outputs of the segmentation network G. To be specific, a is defined for pixel i as follows:

$$a_{ik} = \frac{\exp(\frac{z_{ik}}{T})}{\sum_{j=1}^{K} \exp(\frac{z_{ij}}{T})}, \tag{6}$$

where z_{ik} is the logit for class k from G and T is a hyperparameter (temperature). We denote the label by $[a^{(s)}; a^{(t)}]$ below. Then, we minimize the following with respect to G while freezing D.

$$\mathcal{L}_{adv}^{cra} = - \sum_{i=1}^{H \times W} \sum_{k=1}^{K} a_{ik}^{(t)} \overline{m}_i \log P(d = 0, c = k \mid f_i). \tag{7}$$

The mask $\overline{m}_i$ ensures the sum is taken over the untrusted pixels. We then train D to classify the input feature f_i as trusted or untrusted regions as accurately as possible. This is done by minimizing

$$\begin{aligned} \mathcal{L}_{D}^{cra} = & - \sum_{i=1}^{H \times W} \sum_{k=1}^{K} a_{ik}^{(s)} m_i \log P(d = 0, c = k \mid f_i) \\ & - \sum_{i=1}^{H \times W} \sum_{k=1}^{K} a_{ik}^{(t)} \overline{m}_i \log P(d = 1, c = k \mid f_i). \end{aligned} \tag{8}$$

As above, m_i and $\overline{m}_i$ ensure the two sums taken over trusted and untrusted pixels, respectively.

In summary, after obtaining the trusted and untrusted regions we perform the following with a single minibatch and repeat it for a certain number of iterations:

1. Update G using $\mathcal{L}_{seg}^{cra}$ of (5) with the pseudo labels (4) on the trusted regions.
2. Compute the domain encoding label with class probabilities by (6) and obtain $[0; a^{(t)}]$ for the untrusted region data and $[a^{(s)}; 0]$ for the trusted region data.
3. Update G using $\mathcal{L}_{adv}^{cra}$ of (7).
4. Update D using $\mathcal{L}_{D}^{cra}$ of (8).

4. Experiments

4.1. Datasets

We evaluate our method on three scenarios of domain adaptation from a synthetic to a real dataset. For the source domain dataset, we consider either of GTA5 (Richter et al., 2016), SYNTHIA (Ros et al., 2016), and Synscapes (Wrenninge and Unger, 2018). For the target domain dataset, we consider Cityscapes (Cordts et al., 2016).

Cityscapes (Cordts et al., 2016) This is an urban scene dataset consisting of data collected from the real world, which contains 19 categories. It provides 2975 images for training, 500 images for validation, and 1525 images for testing. For the use of these data, we follow previous studies (Vu et al., 2019a; Tsai et al., 2018; Wang et al., 2020). We report the performance on the validation set below unless otherwise noted. We also report the results on the test set in Section 4.10.

GTA5 (Richter et al., 2016) This is a synthetic dataset generated from a video game. It contains 24,966 images with segmentation labels and shares the same 19 classes as Cityscapes. We use all the images as the source training data.

SYNTHIA (Ros et al., 2016) This is a synthetic dataset that mainly contains urban scene samples. In our experiments, we use the SYNTHIA-RAND-CITYSCAPES subset as our source domain training data, which shares 16 classes with Cityscapes and has 9400 images with segmentation labels.

Synscapes (Wrenninge and Unger, 2018) This is a photorealistic synthetic dataset for street scene parsing. It shares the same 19 classes as Cityscapes and contains 25,000 images with segmentation labels.

4.2. Experimental configuration

For the design of the segmentation network G and the discriminator D, we followed previous studies. Specifically, we use DeepLabv2 (Chen et al., 2017) with two different backbones (i.e., feature extraction networks): VGG-16 (Simonyan and Zisserman, 2015) and ResNet101 (He et al., 2016) for G. Both backbones are pretrained on ImageNet (Deng et al., 2009). For D, we use a network consisting of three convolution layers. We train G and D as explained in Section 3.3. First, we train G on the source domain and then apply a base UDA method, following the experimental setting of the base method. Finally, we apply the proposed CRA for 40k iterations.

We select existing methods for the base UDA method, including the current state-of-the-art methods: ASN (Tsai et al., 2018), ADVENT (Vu et al., 2019a), IntraDA (Pan et al., 2020), FADA (Wang et al., 2020), IAST (Mei et al., 2020), ProDA (Zhang et al., 2021), DAFormer (Hoyer et al., 2022a), and MIC (Hoyer et al., 2023). The results of combining a base method with the proposed CRA will be denoted as "(the base)+CRA". For instance, ProDA+CRA refers to the combination of ProDA with the proposed CRA.

We use the SGD optimizer for the training of G with momentum $= 0.9$ and weight decay $= 10^{-4}$. The learning rate follows polynomial decay from 2.5×10^{-4} with power of 0.9. We use the Adam optimizer for the training of D; the initial learning rate is set to 10^{-4}, and $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We use the polynomial decay of learning rate.

For evaluation, we follow the standard method for semantic segmentation (Chen et al., 2017); we use the intersection-over-union (IoU) (Everingham et al., 2015) as our metric. We report per-class IoU and mean IoU over all classes.

4.3. Main results on different UDA scenarios

4.3.1. GTA5 → Cityscapes

Table 2 shows the results for the GTA5 → Cityscapes adaptation. It is first observed that combining CRA with any of the base methods tested led to an enhancement in the performance. The extent of improvement varies for each base method, ranging from 0.7pp to 3.4pp. The stronger baselines tend to show smaller improvements, which is reasonable as they may be closer to the maximum achievable accuracy under the given conditions. However, it is noteworthy that CRA improves even the state-of-the-art method, MIC (Hoyer et al., 2023); MIC+CRA achieves 59.2 mean IoU, which is an improvement of 1.0pp over MIC. These results confirm the effectiveness of the proposed approach.

Fig. 3 shows a few examples of the results of FADA+CRA. It is observed that while the base method (i.e., FADA) fails in most of the image regions classified as untrusted based on entropy, the application of CRA consistently leads to improved segmentation results.

Table 2 displays the results obtained from the same baselines (ASN, ADVENT, and FADA) but with different backbone networks (VGG16 and ResNet101). It can be observed that the improvements tend to be more significant for VGG16 compared to ResNet101, i.e., ASN (3.1pp vs. 1.0pp), ADVENT (3.2pp vs. 2.9p), and FADA (3.4pp vs. 2.1pp).

4.3.2. SYNTHIA → Cityscapes

Table 3 presents the results for the SYNTHIA → Cityscapes adaptation. Similar to the GTA5 → Cityscapes, combining CRA with any baseline results in performance improvement, ranging from 1.2pp to 3.3pp in the mIoU_{13} metric. Additionally, MIC+CRA achieves mIoU_{16} and mIoU_{13} scores of 57.2 and 64.7, respectively, representing improvements of 1.3pp and 1.2pp over MIC.

Table 2
Results of GTA5 → Cityscapes. '*+CRA' in the 'method' column represents the combination of a CDA method and the proposed CRA. The column 'B' indicates backbones; V and R means VGG-16 and ResNet101, respectively. '(Δ)' in the last column indicates the improvement from the base CDA. The best result is highlighted for each class.

Method	B	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU (Δ)
Source Only	V	35.4	13.2	72.1	16.7	11.6	20.7	22.5	13.1	76.0	7.6	66.1	41.1	19.0	69.8	15.2	16.3	0.0	16.2	4.7	28.3
ASN (Tsai et al., 2018)	V	87.3	29.8	78.6	21.1	18.2	22.5	21.5	11.0	79.7	29.6	71.3	46.8	6.5	80.1	23.0	26.9	0.0	10.6	0.3	35.0
ASN+CRA	V	91.0	47.9	80.2	28.9	12.0	30.2	23.3	11.5	80.7	34.3	75.0	45.6	17.0	79.4	19.9	27.1	0.0	11.9	7.6	38.1 (+3.1)
ADVENT (Vu et al., 2019a)	V	86.9	28.7	78.7	28.5	25.2	17.1	20.3	10.9	80.0	26.4	70.2	47.1	8.4	81.5	26.0	17.2	18.9	11.7	1.6	36.1
ADVENT+CRA	V	92.8	55.9	80.6	23.4	18.4	31.1	23.3	4.5	82.7	37.1	80.3	48.2	17.8	81.6	19.4	21.0	0.0	9.4	18.7	39.3 (+3.2)
FADA (Wang et al., 2020)	V	92.3	51.1	83.7	33.1	29.1	28.5	28.0	21.0	82.6	32.6	85.3	55.2	28.8	83.5	24.4	37.4	0.0	21.1	15.2	43.8
FADA+CRA	V	93.3	59.3	84.6	24.8	26.1	36.2	33.5	30.2	84.2	38.4	85.6	60.4	29.6	85.0	24.7	39.0	20.4	24.3	17.5	47.2 (+3.4)
Source Only	R	65.0	16.1	68.7	18.6	16.8	21.3	31.4	11.2	83.0	22.0	78.0	54.4	33.8	73.9	12.7	30.7	13.7	28.1	19.7	36.8
ASN (Tsai et al., 2018)	R	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
ASN+CRA	R	87.7	33.5	81.6	33.3	23.7	22.4	26.1	16.0	83.6	38.5	72.7	52.7	21.2	84.2	44.9	43.0	0.0	26.2	33.0	43.4 (+1.0)
ADVENT (Vu et al., 2019a)	R	89.9	36.5	81.6	29.2	25.2	28.5	32.3	22.4	83.9	34.0	77.1	57.4	27.9	83.7	29.4	39.1	1.5	28.4	23.3	43.8
ADVENT+CRA	R	90.0	39.9	83.5	33.3	27.5	28.7	35.0	27.6	85.2	37.2	80.0	57.9	29.4	85.1	39.3	44.2	0.0	26.4	37.6	46.7 (+2.9)
IntraDA (Pan et al., 2020)	R	90.6	36.1	82.6	29.5	21.3	27.6	31.4	23.1	85.2	39.3	80.2	59.3	29.4	86.4	33.6	53.9	0.0	32.7	37.6	46.3
IntraDA+CRA	R	90.7	38.7	83.4	30.8	23.7	27.0	32.2	28.9	85.2	38.8	81.8	59.0	28.9	86.0	32.7	49.8	0.0	37.2	39.1	47.0 (+0.7)
FADA (Wang et al., 2020)	R	91.0	50.6	86.0	43.4	29.8	36.8	43.4	25.0	86.8	38.3	87.4	64.0	38.0	85.2	31.6	46.1	6.5	25.4	37.1	50.1
FADA+CRA	R	91.6	53.6	85.5	42.6	18.7	34.8	36.3	18.9	87.8	45.0	89.0	66.2	39.0	87.6	42.3	51.3	14.9	42.0	44.6	52.2 (+2.1)
IAST (Mei et al., 2020)	R	94.1	58.8	85.4	39.7	29.2	25.1	43.1	34.2	84.8	34.6	88.7	62.7	30.3	87.6	42.3	50.3	**24.7**	35.2	40.2	52.2
IAST+CRA	R	93.3	57.8	87.2	42.1	29.6	40.9	50.8	51.0	86.0	28.2	87.7	67.8	34.4	87.5	29.8	46.3	14.2	40.8	52.7	54.1 (+1.9)
ProDA (Zhang et al., 2021)	R	87.8	56.0	79.7	46.3	44.8	**45.6**	53.5	53.5	88.6	45.2	82.1	70.7	39.2	88.8	45.5	59.4	1.0	48.9	56.4	57.5
ProDA+CRA	R	89.4	60.0	81.0	**49.2**	**44.8**	45.5	53.6	55.0	**89.4**	**51.9**	85.6	72.3	40.8	88.5	44.3	53.4	0.0	**51.7**	**57.9**	58.6 (+1.1)
DAFormer (Hoyer et al., 2022a)	R	95.9	71.4	88.0	35.3	30.7	41.5	51.8	54.3	87.2	47.4	86.9	67.8	30.2	89.8	48.6	47.9	0.0	24.7	41.0	54.7
DAFormer+CRA	R	**96.2**	**73.5**	88.7	36.9	29.0	44.2	**54.4**	**61.9**	88.1	48.2	87.2	70.0	27.5	**90.3**	43.3	46.4	0.0	28.0	41.5	55.5(+0.8)
MIC (Hoyer et al., 2023)	R	95.2	67.9	88.7	39.7	37.6	39.3	51.7	56.5	88.5	47.5	89.5	70.3	46.4	88.8	**50.2**	**61.8**	6.6	28.1	51.3	58.2
MIC+CRA	R	96.1	71.7	**89.0**	37.5	39.7	42.1	53.3	59.0	89.3	46.3	**89.7**	**72.3**	**49.9**	89.4	42.0	60.8	12.2	28.3	56.7	**59.2** (+1.0)

Table 3
Results of SYNTHIA → Cityscapes; $mIoU_{16}$ and $mIoU_{13}$ denote the mIoU scores across 16 and 13 classes respectively. '*+CRA' in the 'method' column represents the combination of a CDA method and the proposed CRA. The column 'B' indicates backbones; V and R means VGG-16 and ResNet101, respectively. '(Δ)' in the last column indicates the improvement from the base CDA. The best result is highlighted for each class. For ASN, three categories are excluded following the original paper.

Method	B	road	sidewalk	building	wall	fence	pole	light	sign	veg	sky	person	rider	car	bus	mbike	bike	$mIoU_{16}$ (Δ)	$mIoU_{13}$ (Δ)
Source Only	V	10.0	14.7	52.4	4.2	0.1	20.9	3.5	6.5	74.3	77.5	44.9	4.9	64.0	21.6	4.2	6.4	25.6	29.6
ASN (Tsai et al., 2018)	V	78.9	29.2	75.5	-	-	-	0.1	4.8	72.6	76.7	43.4	8.8	71.1	16.0	3.6	8.4	-	37.6
ASN+CRA	V	81.6	35.8	76.8	-	-	-	0.0	2.2	77.9	78.1	43.7	9.6	61.8	22.7	2.7	20.5	-	39.3 (+1.7)
ADVENT (Vu et al., 2019a)	V	67.9	29.4	71.9	6.3	0.3	19.9	0.6	2.6	74.9	74.9	35.4	9.6	67.8	21.4	4.1	15.5	31.4	36.6
ADVENT+CRA	V	75.7	33.8	74.7	5.5	0.0	21.9	0.0	2.4	78.7	78.4	42.8	10.4	61.4	22.9	3.2	14.3	32.9 (+1.5)	38.4 (+1.8)
FADA (Wang et al., 2020)	V	80.4	35.9	80.9	2.5	0.3	30.4	7.9	22.3	81.8	83.6	48.9	16.8	77.7	31.1	13.5	17.9	39.5	46.0
FADA+CRA	V	83.7	38.7	79.4	0.3	0.6	30.6	0.0	8.5	81.8	78.1	58.7	17.4	80.7	32.8	14.6	45.3	40.7 (+1.2)	47.7 (+1.7)
Source Only	R	55.6	23.8	74.6	9.2	0.2	24.4	6.1	12.1	74.8	79.0	55.3	19.1	39.6	23.3	13.7	25.0	33.5	38.6
ASN (Tsai et al., 2018)	R	84.3	42.7	77.5	-	-	-	4.7	7.0	77.9	82.5	54.3	21.0	72.3	32.2	18.9	32.3	-	46.7
ASN+CRA	R	84.7	36.6	78.5	-	-	-	12.9	14.5	79.1	81.7	57.3	25.9	74.2	20.0	30.9	44.5	-	49.3 (+2.6)
ADVENT (Vu et al., 2019a)	R	85.6	42.2	79.7	8.7	0.4	25.9	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	41.2	48.0
ADVENT+CRA	R	89.6	47.3	80.9	10.6	0.1	32.0	9.7	15.5	82.2	82.8	60.7	25.9	76.9	28.3	16.0	50.3	44.3 (+3.1)	51.2 (+3.2)
IntraDA (Pan et al., 2020)	R	84.3	37.7	79.5	5.3	0.4	24.9	9.2	8.4	80.0	84.1	57.2	23.0	78.0	38.1	20.3	36.5	41.7	48.9
IntraDA+CRA	R	**90.1**	44.7	80.8	11.3	0.0	28.1	6.3	12.3	82.1	80.4	58.6	24.5	85.0	41.1	21.3	52.3	44.9 (+3.2)	52.2 (+3.3)
FADA (Wang et al., 2020)	R	84.5	40.1	83.1	4.8	0.0	34.3	20.1	27.2	84.8	84.0	53.5	22.6	85.4	43.7	26.8	27.8	45.2	52.5
FADA+CRA	R	88.8	**48.3**	83.4	12.6	0.4	36.4	8.1	22.2	86.6	80.9	64.8	22.4	87.7	45.5	29.9	46.4	47.8 (+2.6)	55.0 (+2.5)
IAST (Mei et al., 2020)	R	81.9	41.5	83.3	17.7	4.6	32.3	30.9	28.8	83.4	85.0	65.5	30.8	86.5	38.2	33.1	52.7	49.8	57.0
IAST+CRA	R	75.8	33.3	85.1	36.7	0.0	42.5	46.7	37.2	85.4	83.7	68.5	32.0	87.8	44.9	40.7	53.8	53.4 (+3.6)	59.6 (+2.6)
ProDA (Zhang et al., 2021)	R	87.8	45.7	84.6	37.1	0.6	**44.0**	54.6	37.0	88.1	84.4	74.2	24.3	88.2	51.1	40.5	45.6	55.5	62.0
ProDA+CRA	R	85.6	44.2	82.7	**38.6**	0.4	43.5	**55.9**	42.8	**87.4**	85.8	**75.8**	27.4	**89.1**	**54.8**	46.6	49.8	56.9 (+1.4)	63.7 (+1.7)
DAFormer (Hoyer et al., 2022a)	R	60.8	16.9	84.3	15.2	3.5	37.1	46.0	47.0	85.0	85.7	70.7	42.9	87.0	44.3	43.7	57.4	51.7	59.4
DAFormer+CRA	R	64.2	18.8	84.9	18.9	4.7	38.4	48.9	50.5	85.6	**86.8**	72.8	44.2	87.7	44.8	46.8	63.1	53.8 (+2.1)	61.4 (+2.0)
MIC (Hoyer et al., 2023)	R	66.5	26.6	86.2	26.7	3.7	39.4	48.6	53.0	85.0	83.6	71.6	47.8	88.1	50.0	**55.1**	63.1	55.9	63.5
MIC+CRA	R	72.4	30.2	**86.8**	26.7	**4.8**	41.8	49.9	**54.7**	84.9	84.4	73.3	**48.0**	88.6	50.6	51.3	**66.2**	**57.2** (+1.3)	**64.7** (+1.2)

Regarding the effects of the backbone networks, we have a different result here. The use of VGG16 as a backbone network tends to exhibit lower improvements compared to ResNet101 for the same baselines, i.e., ASN (1.7pp vs. 2.6pp), ADVENT (1.8pp vs. 3.2pp), and FADA (1.7pp vs. 2.5pp).

4.3.3. Synscapes → Cityscapes

Table 4 shows the results for the Synscapes → Cityscapes adaptation. The main observation is the same; CRA improves all the baselines. The improvement ranges from 0.7pp to 3.7pp. MIC+CRA yields 60.6 mIoU, which improves MIC by 2.1pp; this is larger than the above two adaptation scenarios. These confirm the effectiveness of CRA once again.

Regarding the effects of the backbone network, the results show a similar tendency to GTA5 → Cityscapes; the improvements tend to be larger for VGG16 than ResNet101.

4.4. Performance on different backbones

As shown in the above experiments (Tables 2–4), the combination of CRA with the same base methods produces different degrees of improvement with different backbone networks. So far, we have used VGG16 and ResNet101 as backbones, rigorously, VGG16-based and ResNet101-based DeepLabv2 models as the base segmentation network. We consider ResNet101 as the primary one since it is commonly used in existing studies, and we include VGG16 since it has been used in some earlier studies in addition to ResNet101.

Recent studies on UDA for semantic segmentation (Hoyer et al., 2022a, 2023) have shown that switching the base segmentation network to SegFormer (Xie et al., 2021) (Transformer-based segmentation network) improves performance. This raises the natural question of whether CRA is also effective to the Transformer-based baselines.

To answer this question, we conducts additional experiments by incorporating CRA into DAFormer (Hoyer et al., 2022a). Table 5 shows the results, where DAFormer+CRA outperforms the original DAFormer in all three domain adaptation scenarios, with performance improvements of approximately 1.1pp. (Note that DAFormer (and MIC) reported in Tables 2–3 employ ResNet101 as backbones.) This result further confirms the versatility of CRA, as it can be combined with any UDA method, leading to a certain degree of improvement.

4.5. Comparison with entropy minimization

CRA aligns the feature distributions of the untrusted (i.e., high-entropy) regions and the trusted (low-entropy) regions. As it will effectively reduce the areas of the untrusted regions, the closest approach is the traditional entropy minimization. To evaluate the effect of our CRA, we conducts an experiment on the GTA5 → Cityscapes setting and compare CRA with the entropy minimization and a few baseline methods. For the base UDA method, we chose FADA without its two optional steps (i.e., self-distillation and multi-scale testing) (Wang et al., 2020). We first apply it to the target domain images, splitting the image into the trusted and untrusted regions. We then apply two methods instead of CRA. One is the entropy minimization, where we use a loss function minimizing the entropy of untrusted regions along with the standard cross entropy loss with the pseudo labels on the trusted regions. The other is to retrain the base model using the pseudo labels on the trusted regions (Zou et al., 2018, 2019). As seen in Table 6, CRA works better than the baseline methods with a good margin.

4.6. Ablation test and effects of optional steps

We conduct an ablation test for the combination of FADA (Wang et al., 2020) and the proposed CRA. As FADA employs two optional steps, self-distillation (SD) and a multi-scale test (MST), we also examine their effectiveness. We use the trained FADA+CRA model to generate pseudo labels for the target domain training data and fine-tune G for additional 20k iterations for self-distillation. For the multi-scale test, we scale each image by the factors of $\{0.7, 1.0, 1.3\}$, feed the scaled images separately to G, and fuse the resulting score maps by averaging at each position. Please note that we do not use these optional steps on methods other than FADA to improve performance.

Table 7 shows the results. They indicate that CRA alone does improve performance (36.8 → 41.7) but is inferior to CDA alone (46.9); their combination leads to a large improvement (50.4) over CRA alone. The employment of SD brings about a good amount of further improvement. MST also helps, but its effect is small.

In the FADA (Wang et al., 2020), the training process (P_1) is as follows: training on the source domain → FADA training → self-distillation on the target domain (SD). Among these steps, the second step is the core method, while the third step, SD, is an optional step for further performance improvement. Therefore, when combining our proposed CRA with FADA, our training process (P_2) is: training on the source domain → FADA training → CRA training → SD, with SD remaining as an optional step placed at the end. We also tested another process P_3, which is: training on the source domain → FADA training → SD → CRA training. Compared to P_2's mIoU score of 52.0, P_3's performance is lower at 51.0, but still 1.9pp higher than P_1's 49.2. This result further demonstrates the effectiveness of our proposed method.

4.7. Ablation test of entropy scaling

As there is a class imbalance issue in segmentation data, some classes that occupy a small percentage in the data are likely to yield high entropy values of the prediction probability. This could hurt the accuracy of the classes during CRA training. To avoid this issue, we calculate the percentage of each class in the pseudo labels $\hat{y}^{(t)}$, and define the classes that account for less than 1% of all the pixels as rare classes. We then adjust their entropy values by multiplying them by a scalar F_s during the CRA training. The weighting strategy is applied only to the long-tail classes. Experiments are conducted on the GTA5 → Cityscapes setting to verify the effect of the scaling. Table 8 shows that the entropy scaling can make our method performs better, and we can obtain the best performance with $F_s = 0.5$.

For the three UDA tasks considered in the paper, their target domain is Cityscapes, so we employed the same set of hyper-parameters for all tasks, which is a common practice in previous research (Pan et al., 2020; Wang et al., 2020; Zhang et al., 2021). We believe that a more refined parameter tuning for each UDA task and specific UDA model can further boost performance, yet this is rather tricky and brings a significant workload. Therefore, we opted to follow the precedent, applying the parameters of GTA5 → Cityscapes to all.

4.8. Visualization of feature spaces

To evaluate the effectiveness of our proposed approach from a different perspective, we visualize the changes in the feature space at each training stage. Following Zhang et al. (2021), we show a low-dimensional mapping of the feature space using UMAP (McInnes et al., 2018). Specifically, we plot the features of image pixels belonging to the same-class regions at three stages: (i) the initial stage after training the model on source-domain data in a supervised manner, (ii) after applying a base UDA method, and (iii) after applying CRA. We choose the scenario of GTA → Cityscapes and FADA as the base method; we do not employ self-distillation or multi-scale training for simplicity. Fig. 4 displays the results, where dots in a different color indicate pixels belonging to the same ground-truth class. We observe that initially non-separated features become separated after applying the base UDA method and further separated after applying CRA.

Table 4
Results of Synscapes → Cityscapes.'*+CRA' in the 'method' column represents the combination of a CDA method and the proposed CRA. The column 'B' indicates backbones; V and R means VGG-16 and ResNet101, respectively. '(Δ)' in the last column indicates the improvement from the base CDA. The best result is highlighted for each class.

Method	B	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU (Δ)
Source Only	V	89.5	42.8	74.0	23.1	18.7	25.6	20.3	14.6	78.6	27.0	78.6	49.1	26.6	77.6	10.6	15.3	5.5	9.8	31.6	37.8
ASN (Tsai et al., 2018)	V	89.9	45.9	78.5	16.2	22.7	25.3	24.0	18.9	81.2	30.4	81.4	46.2	29.2	79.9	7.6	14.6	11.3	20.1	35.4	39.9
ASN+CRA	V	92.6	50.1	80.8	23.6	26.9	33.0	29.9	23.1	83.4	24.2	79.9	53.9	32.4	80.4	13.9	29.1	9.7	11.7	42.3	43.2 (+3.3)
ADVENT (Vu et al., 2019a)	V	91.1	50.8	78.0	15.5	19.1	25.3	24.1	20.3	80.7	27.7	79.4	49.3	30.1	79.6	8.7	15.3	6.8	21.1	37.0	40.0
ADVENT+CRA	V	92.7	50.8	80.6	20.7	24.7	34.2	28.9	26.9	83.2	24.8	80.7	57.3	34.0	78.9	17.6	27.9	9.8	11.4	45.5	43.7(+3.7)
FADA (Wang et al., 2020)	V	91.6	45.7	77.4	28.4	19.8	27.5	22.9	21.5	80.7	15.8	81.8	47.0	28.7	78.7	12.0	17.4	11.8	13.8	39.2	40.1
FADA+CRA	V	92.3	51.0	78.7	22.3	26.1	35.2	29.0	24.7	83.1	22.9	81.8	57.6	34.3	81.9	10.8	17.6	6.9	14.2	46.2	43.0 (+2.9)
Source Only	R	81.8	40.6	76.1	23.3	16.8	36.9	36.8	40.1	83.0	34.8	84.9	59.9	37.7	78.4	20.4	20.5	7.8	27.3	52.5	45.3
ASN (Tsai et al., 2018)	R	93.4	55.9	82.8	30.6	27.5	36.9	38.7	40.4	84.5	33.1	87.6	56.9	37.1	86.0	37.7	43.6	20.2	27.7	57.3	51.5
ASN+CRA	R	92.6	54.4	83.7	30.4	27.4	38.7	34.6	44.1	85.1	37.4	87.8	63.7	39.9	86.4	38.5	42.7	15.8	31.3	57.4	52.2 (+0.7)
ADVENT (Vu et al., 2019a)	R	93.4	57.6	83.7	29.4	27.4	37.1	40.3	42.4	85.6	30.2	88.0	58.4	35.7	86.7	34.5	45.8	21.5	24.2	55.7	51.5
ADVENT+CRA	R	92.8	57.3	84.4	28.2	27.8	38.8	37.4	45.7	86.1	35.5	87.5	64.1	37.8	87.1	35.0	44.8	20.8	28.7	58.0	52.5 (+1.0)
IntraDA (Pan et al., 2020)	R	93.3	57.7	84.7	23.5	29.8	36.8	41.0	41.3	86.1	34.9	88.3	60.0	36.8	87.2	40.9	48.4	15.0	33.7	58.8	52.5
IntraDA+CRA	R	93.2	57.7	84.8	27.6	31.9	36.8	40.5	42.2	86.2	37.3	87.9	61.4	38.6	87.5	**41.5**	49.2	15.6	33.2	58.7	53.3 (+0.8)
FADA (Wang et al., 2020)	R	94.1	58.4	84.4	33.7	34.7	38.4	44.0	43.2	84.7	40.6	89.2	62.9	38.8	87.0	29.3	40.2	18.2	33.1	55.8	53.2
FADA+CRA	R	94.2	58.9	85.8	33.9	37.8	41.7	47.6	46.3	85.6	45.4	90.3	66.8	40.0	87.1	31.5	43.5	18.9	37.6	59.5	55.3 (+2.1)
IAST (Mei et al., 2020)	R	94.0	64.4	85.7	34.2	35.2	47.4	51.0	61.1	88.1	42.0	90.5	70.4	35.3	86.7	28.3	37.0	23.1	32.8	52.5	55.8
IAST+CRA	R	**94.9**	68.0	**87.2**	**37.4**	39.5	48.5	50.6	64.3	88.2	39.9	88.9	72.7	36.4	86.2	29.8	37.7	18.2	31.6	54.4	56.5 (+0.7)
ProDA (Zhang et al., 2021)	R	90.7	43.1	85.3	29.5	32.8	50.1	61.8	**64.5**	89.4	48.6	89.1	76.3	47.0	88.9	36.5	54.4	**34.8**	44.4	61.0	59.4
ProDA+CRA	R	90.1	42.9	85.0	33.4	36.7	**52.2**	**63.7**	64.0	**90.1**	48.5	89.6	**77.6**	49.4	89.3	36.1	**56.6**	32.9	44.3	61.3	60.2 (+0.8)
DAFormer (Hoyer et al., 2022a)	R	79.4	37.5	71.8	26.0	17.9	46.5	57.0	58.4	81.6	45.7	91.3	71.4	45.7	90.1	18.7	44.6	3.5	38.2	68.9	52.3
DAFormer+CRA	R	78.6	31.3	73.3	24.1	19.3	47.6	60.9	63.3	82.2	46.7	**91.9**	74.2	50.4	**90.7**	23.2	46.9	1.8	49.6	**71.0**	54.1 (+1.8)
MIC (Hoyer et al., 2023)	R	91.2	73.0	82.3	18.8	37.4	47.1	58.9	54.5	89.0	50.9	91.0	72.5	48.5	89.3	20.4	41.3	25.5	51.4	68.8	58.5
MIC+CRA	R	93.2	**75.1**	83.3	22.5	**41.3**	47.5	61.6	55.8	89.9	**53.0**	89.2	76.0	**51.6**	89.7	22.6	42.0	31.8	**55.0**	70.2	**60.6** (+2.1)

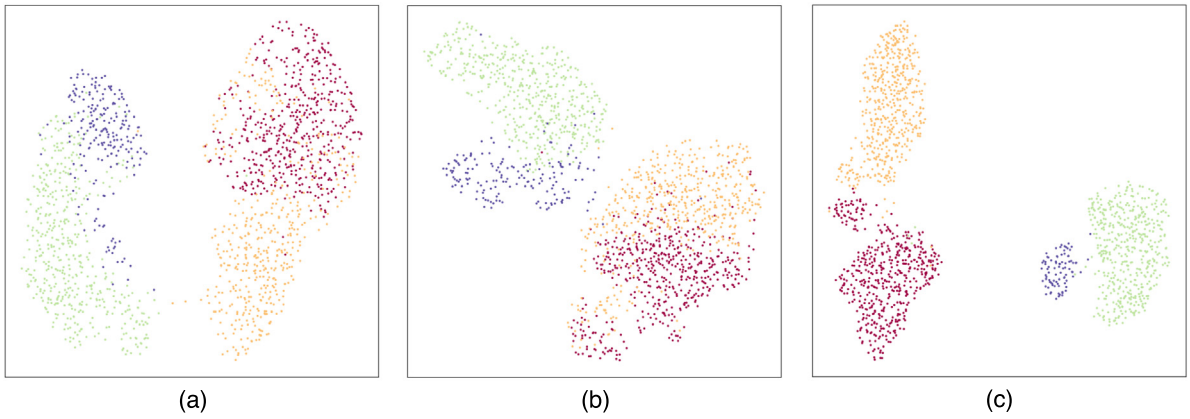


Fig. 4. Visualization of feature space at different stages of the training process. (a) The initial stage after the model is trained using supervised learning on the source-domain data. (b) Results after applying the base method, i.e., FADA (Wang et al., 2020). (c) Results after applying CRA. The evaluation was performed on the GTA5 → Cityscapes dataset, and to ensure clarity, we selected four categories: roads (red), sidewalks (orange), cars (green), and buses (purple). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Performance of CRA with DAFormer (Hoyer et al., 2022a). The mIoU score of SYNTHIA → Cityscapes is evaluated based on 16 classes.

Source Domain	DAFormer	DAFormer+CRA
GTA5	68.3	69.5 (+1.2)
SYNTHIA	60.9	62.0 (+1.1)
Synscapes	61.8	62.9 (+1.1)

Table 6
Comparison with entropy minimization on GTA5 → Cityscape.

Method	mIoU
CDA (FADA w/o options)	46.9
Self training on trusted regions	47.5
Entropy minimization	48.4
CRA	**50.4**

Table 7
Ablation test results of FADA+CRA with optional steps, i.e., self-distillation (SD) and multi-scale test (MST), on GTA5 → Cityscapes task. CDA indicates FADA without the optional steps.

Source Only	CDA	CRA	SD	MST	mIoU
✓					36.8
✓	✓				46.9
✓		✓			41.7
✓	✓		✓		49.2
✓	✓		✓	✓	50.1
✓	✓	✓			50.4
✓	✓	✓	✓		52.0
✓	✓	✓	✓	✓	**52.2**

Table 8
Effects of the scaling factor F_s for rare classes. Results on GTA5 → Cityscapes obtained by FADA+CRA with ResNet101 backbone.

F_s	0.25	0.5	0.75	1.0
mIoU	49.7	**50.4**	49.4	48.6

4.9. Selection of the entropy threshold λ

In Section 3.3.2, we describe our method for classifying image pixels into trusted and untrusted categories by comparing the entropy of the predicted class probability with a threshold value, λ. While we have used a simple approach to choose a fixed value for λ, there may be a more effective approach that can improve the overall impact of our method. To explore this possibility, we conduct an experimental evaluation of a more flexible method that increases λ linearly from an initial value of λ_I to a final value of λ_F as the training progresses. The rationale behind this approach is to be more cautious in the early stages of training when the model is not yet well-trained, and gradually increase the number of pixels that we can trust as training proceeds.

Table 9
Results of CRA with a scheduler of the entropy threshold λ on GTA → Cityscapes. FADA without self-distillation and multi-scale test is used as the base method.

λ_I \ λ_F	0.005	0.01	0.015	0.02	0.025
0.0005	48.2	49.7	49.0	49.2	48.0
0.001	49.0	49.1	49.7	50.0	49.1
0.0015	49.2	49.7	48.5	48.5	48.8
0.002	49.2	49.4	49.1	50.7	48.7
0.0025	48.9	50.4	49.7	48.8	49.7

In the experiment, we choose FADA as the base method and test its combination with CRA on GTA5 → Cityscapes. For the sake of simplicity, we do not use self-distillation or multi-scale test; this corresponds to the sixth row, yielding 50.4 mIoU, in Table 7. The results obtained by the new scheduling method with different combinations of λ_I and λ_F are shown in Table 9. While the above new method occasionally achieves better performance, the improvements are only marginal. Thus, we conclude that our method explained in Section 3.3.2 is a good choice, given the cost of hyper-parameter tuning.

4.10. Evaluation with the Cityscapes test set

The Cityscapes dataset is primarily used as the target domain dataset in existing studies. The dataset consists of training, validation, and test subsets. The ground truth labels for the test subset are unavailable, and the evaluation of results on it needs to post them to the official server (Chen et al., 2017; Fu et al., 2019; Zhang et al., 2018). Thus, a common practice of the existing studies is to evaluate methods on the validation subset. However, this may make the fair comparison very hard, considering the necessity of choosing hyperparameters also on the validation subset. To cope with this, we evaluate the results for the test subset by the proposed method and the compared methods on the official server with the models of GTA5 → Cityscapes task. Table 10 shows the results, which validate the effectiveness of the proposed approach.

Table 10
Performance evaluation of eight base methods and their CRA extensions on Cityscapes test set. These results are calculated by the official server by uploading predicted segmentation masks.

Method	mIoU (Δ)
ASN (Tsai et al., 2018)	43.6
ASN+CRA	45.9 (+2.3)
ADVENT (Vu et al., 2019a)	45.7
ADVENT+CRA	48.1 (+2.4)
IntraDA (Pan et al., 2020)	47.2
IntraDA+CRA	48.8 (+1.6)
FADA (Wang et al., 2020)	52.4
FADA+CRA	53.8 (+1.4)
IAST (Mei et al., 2020)	54.4
IAST+CRA	55.6 (+1.2)
ProDA (Zhang et al., 2021)	56.9
ProDA+CRA	58.9 (+2.0)
DAFormer (Hoyer et al., 2022a)	55.4
DAFormer+CRA	56.6 (+1.2)
MIC (Hoyer et al., 2023)	59.3
MIC+CRA	**60.4** (+1.1)

5. Summary and conclusion

In this study, we have proposed a method called cross-region adaptation (CRA) for unsupervised domain adaptation in semantic segmentation. The proposed method is designed to improve the performance of any existing UDA approach by reducing residual class-level misalignment of feature distributions between the source and target domains. To achieve this, CRA utilizes a self-training framework to split each target domain image into trusted and untrusted regions and performs adversarial training within the target domain to align their feature distributions.

We have shown the experimental results conducted on adapting three different computer graphics (CG) datasets to Cityscapes. Our approach was combined with existing UDA methods, and the results showed that it consistently improves the performance of the combined method. Furthermore, we compared major UDA methods using the Cityscapes test dataset to ensure a fair comparison without any leakage from training to test data. The results demonstrate that the proposed approach achieves the best performance. Overall, these findings confirm the effectiveness of our proposed approach.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was partly supported by JSPS, Japan KAKENHI Grant Number 20H05952 and 19H01110.

References

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39 (12), 2481–2495.

Bucher, M., Vu, T.-H., Cord, M., Pérez, P., 2021. Handling new target classes in semantic segmentation with domain adaptation. Comput. Vis. Image Underst. 212, 103258.

Chatterjee, B., Poullis, C., 2021. Semantic segmentation from remote sensor data and the exploitation of latent learning for classification of auxiliary tasks. Comput. Vis. Image Underst. 210, 103251.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2017. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. 40 (4), 834–848.

Chen, C., Xie, W., Huang, W., Rong, Y., Ding, X., Huang, Y., Xu, T., Huang, J., 2019. Progressive feature alignment for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, pp. 627–636.

Cheng, Y., Wei, F., Bao, J., Chen, D., Wen, F., Zhang, W., 2021. Dual path learning for domain adaptation of semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 9082–9091.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, pp. 3213–3223.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. ImageNet: a large-scale hierarchical image database. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 248–255.

Dong, J., Cong, Y., Sun, G., Zhong, B., Xu, X., 2020. What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 4022–4031.

Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2015. The PASCAL visual object classes challenge: A retrospective. Int. J. Comput. Vis. 111 (1), 98–136.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H., 2019. Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 3146–3154.

Gong, R., Li, W., Chen, Y., Gool, L.V., 2019. DLOW: Domain flow for adaptation and generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, pp. 2477–2486.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 770–778.

Hoyer, L., Dai, D., Van Gool, L., 2022a. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 9924–9935.

Hoyer, L., Dai, D., Van Gool, L., 2022b. HRDA: Context-aware high-resolution domain-adaptive semantic segmentation. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 372–391.

Hoyer, L., Dai, D., Wang, H., Van Gool, L., 2023. MIC: Masked image consistency for context-enhanced domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR.

Lai, X., Tian, Z., Xu, X., Chen, Y., Liu, S., Zhao, H., Wang, L., Jia, J., 2022. DecoupleNet: Decoupled network for domain adaptive semantic segmentation. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 369–387.

Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H., 2018. Diverse image-to-image translation via disentangled representations. In: Proceedings of the European Conference on Computer Vision. ECCV, Springer, pp. 35–51.

Li, Y., Yuan, L., Vasconcelos, N., 2019. Bidirectional learning for domain adaptation of semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, pp. 6936–6945.

Long, M., Cao, Y., Wang, J., Jordan, M., 2015a. Learning transferable features with deep adaptation networks. In: Proceedings of the International Conference on Machine Learning. pp. 97–105.

Long, J., Shelhamer, E., Darrell, T., 2015b. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 3431–3440.

McInnes, L., Healy, J., Melville, J., 2018. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426.

Mei, K., Zhu, C., Zou, J., Zhang, S., 2020. Instance adaptive self-training for unsupervised domain adaptation. In: Proceedings of the European Conference on Computer Vision. ECCV, Springer, pp. 415–430.

Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N., Terzopoulos, D., 2021. Image segmentation using deep learning: A survey. IEEE Trans. Pattern Anal. Mach. Intell.

Pan, F., Shin, I., Rameau, F., Lee, S., Kweon, I.S., 2020. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, pp. 3763–3772.

Pasqualino, G., Furnari, A., Farinella, G.M., 2022. A multi camera unsupervised domain adaptation pipeline for object detection in cultural sites through adversarial learning and self-training. Comput. Vis. Image Underst. 222, 103487.

Richter, S.R., Vineet, V., Roth, S., Koltun, V., 2016. Playing for data: Ground truth from computer games. In: Proceedings of the European Conference on Computer Vision. ECCV, Springer, pp. 102–118.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional networks for biomedical image segmentation. In: Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention. pp. 234–241.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M., 2016. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, pp. 3234–3243.

Saito, K., Watanabe, K., Ushiku, Y., Harada, T., 2018. Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, pp. 3723–3732.

Saporta, A., Vu, T.H., Cord, M., Pérez, P., 2020. ESL: Entropy-guided self-supervised learning for domain adaptation in semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. IEEE.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: Proceedings of the International Conference on Learning Representations.

Sun, B., Saenko, K., 2016. Deep CORAL: Correlation alignment for deep domain adaptation. In: Proceedings of the European Conference on Computer Vision. ECCV, Springer, pp. 443–450.

Tsai, Y.H., Hung, W.C., Schulter, S., Sohn, K., Yang, M.H., Chandraker, M., 2018. Learning to adapt structured output space for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, pp. 7472–7481.

Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P., 2019a. ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, pp. 2517–2526.

Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P., 2019b. DADA: Depth-aware domain adaptation in semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. ICCV, IEEE, pp. 7364–7373.

Wang, H., Shen, T., Zhang, W., Duan, L.Y., Mei, T., 2020. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In: Proceedings of the European Conference on Computer Vision. ECCV, Springer, pp. 642–659.

Wrenninge, M., Unger, J., 2018. Synscapes: A photorealistic synthetic dataset for street scene parsing. arXiv:1810.08705.

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J., 2018. Unified perceptual parsing for scene understanding. In: Proceedings of the European Conference on Computer Vision. ECCV, Springer, pp. 418–434.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P., 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 12077–12090.

Zhang, X., Aliaga, D., 2022. RFCNet: Enhancing urban segmentation using regularization, fusion, and completion. Comput. Vis. Image Underst. 220, 103435.

Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A., 2018. Context encoding for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, pp. 7151–7160.

Zhang, P., Zhang, B., Zhang, T., Chen, D., Wang, Y., Wen, F., 2021. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12414–12424.

Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2017. Pyramid scene parsing network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 2881–2890.

Zheng, Z., Yang, Y., 2020. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. Int. J. Comput. Vis.

Zhou, Q., Feng, Z., Gu, Q., Cheng, G., Lu, X., Shi, J., Ma, L., 2022. Uncertainty-aware consistency regularization for cross-domain semantic segmentation. Comput. Vis. Image Underst. 221, 103448.

Zou, Y., Yu, Z., Kumar, B., Wang, J., 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European Conference on Computer Vision. ECCV, Springer, pp. 289–305.

Zou, Y., Yu, Z., Liu, X., Kumar, B., Wang, J., 2019. Confidence regularized self-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. ICCV, IEEE, pp. 5982–5991.